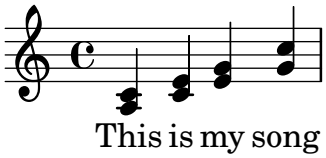# Funcionalidades nuevas de la versión 2.18 desde la 2.16

- Ya no se aplica automáticamente la funcionalidad de ajuste de los trazos de PostScript, sino que se deja a la discreción del dispositivo PostScript (de forma predeterminada, Ghostscript la usa para resoluciones de hasta 150ppp cuando se generan imágenes de matriz de puntos). Cuando se habilita, se emplea (principalmente para las plicas y las barras de compás) un algoritmo de trazado más complejo diseñado para sacar provecho del ajuste de los trazos.

  El ajuste de trazos se puede forzar mediante la especificación de la opción de línea de órdenes '`-dstrokeadjust`' para la instrucción LilyPond. Cuando se generan archivos `PDF`, esto normalmente produce una vista previa de los `PDF` marcadamente mejorada, pero con un tamaño de archivo significativamente más grande. A altas resoluciones, la calidad de la impresión puede no resultar afectada.
- Ahora hay un nuevo tipo de contexto llamado `NullVoice` que, si bien no aparece en la salida impresa, se puede usar para alinear la letra de las canciones. Esto puede ser especialmente conveniente cuando se usa en paralelo con una construcción `\partcombine`.

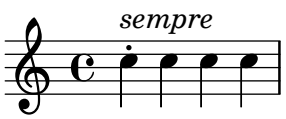
```
soprano = \relative c' { c e g c }
alto = \relative c' { a c e g }
verse = \lyricmode { This is my song }

\score {
  \new Staff <<
    \partcombine \soprano \alto
    \new NullVoice = "aligner" \soprano
    \new Lyrics \lyricsto "aligner" \verse
  >>
  \layout {}
}
```



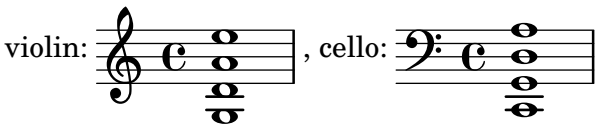

- Pueden colocarse dentro de una sola variable, o ser devueltas por una función de evento, varias articulaciones:

```
sempreStacc = -. ^\markup \italic sempre
\relative { c''4\sempreStacc c c c }
```

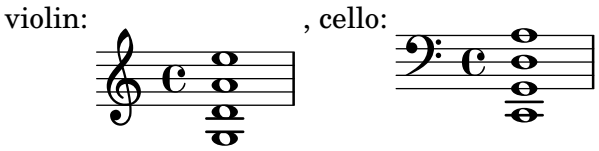



- La línea de base de los elementos de marcado que constituyen partituras se toma actualmente del punto de referencia (normalmente la línea central del pentagrama) del primer sistema de la parte inferior, en lugar de la parte superior del rectángulo circundante. Lo siguente

```
\markup {
  violin: \score { \new Staff { <g d' a' e''>1 }
                   \layout { indent=0 } } ,
  cello: \score { \new Staff { \clef "bass" <c, g, d a> }
                  \layout { indent=0 } }
}
```



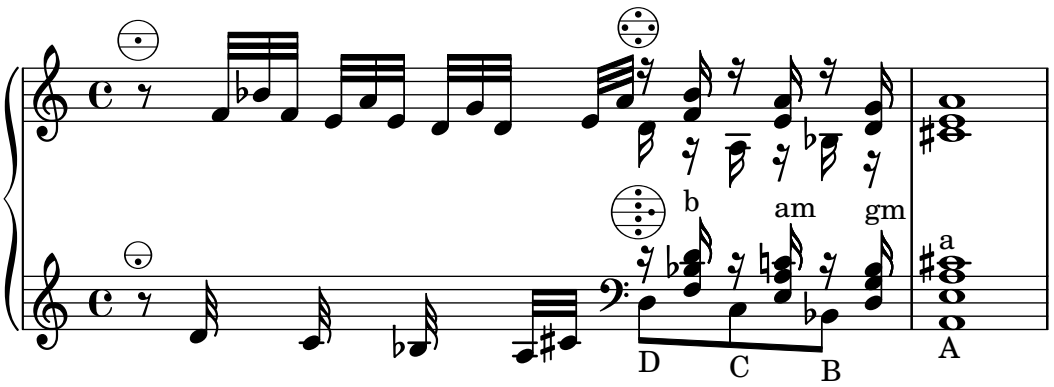

tenía anteriormente el siguiente aspecto:
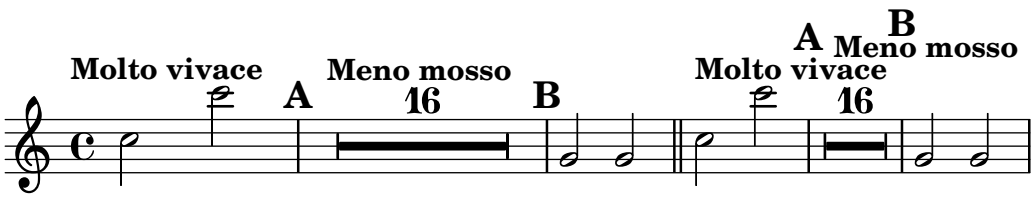



sin que hubiera una manera adecuada de hacer que las dos partituras estuviesen alineadas.

- LilyPond ya no deduce automáticamente un contexto '`\defaultchild`' dentro de una definición de contexto con cláusulas '`\accepts`'. Cualquier definición de contexto que no tenga una definición '`\defaultchild`' explícita o heredada, cuenta como un contexto '`Bottom`' y es candidato para eventos de duraciones y sobreescrituras sin causar la creación implícita de otros contextos. Asegúrese de especificar un '`\defaultchild`' para contextos no-'`Bottom`' al definirlos partiendo de cero.
- Ahora están ampliamente contemplados los símbolos de registros de acordeón tanto para discanto como bajo en el módulo '`scm accreg`', véase Sección "Accordion Registers" in *Referencia de la Notación*.

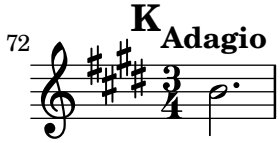
```
#(use-modules (scm accreg))
\new PianoStaff
<<
  \new Staff \relative
  { \clef "treble"  \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    \discant "121"
    << { r16 <f bes> r <e a> r <d g> } \\
       { d r a r bes r } >> |
    <cis e a>1
  }
  \new Staff \relative
  { \clef "treble"  \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef "bass"  \stdBass "Master"
    << { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
         <e a cis>1^"a" } \\
       { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```



- Las nuevas instrucciones `markLengthOn` y `markLengthOff` controlan si se permite la existencia de un espacio horizontal para las letras de ensayo e indicaciones de tempo.




- Ahora las letras de ensayo al principio de una línea se sitúan de forma predeterminada a la derecha de la clave y la armadura. Como en versiones anteriores, la interfaz `break-alignable-interface` controla el comportamiento.




- Ahora se pueden escribir directamente números decimales dentro de la música, sin el prefijo del símbolo de almohadilla. Junto al cambio anterior en la forma de especificar las propiedades de los objetos, el código para modificar la longitud de las plicas ha cambiado de esto:

```
\override Stem #'length = #5.6
e' f' g' a'
```

  a esto:

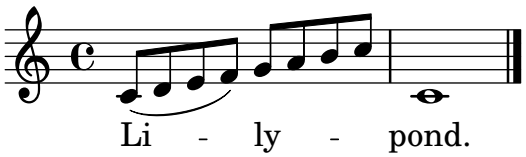
```
\override Stem.length = 5.6
e' f' g' a'
```

  Se debe escribir un dígito a ambos lados del punto; no se permiten valores como `4.` ó `-.3`. Asimismo, no se admiten fracciones decimales dentro de `\chordmode`.

- Se pueden redefinir libremente ciertas abreviaturas como `(`, `)`, `|`, `[`, `]`, `~`, `\(`, `\)` y otras, como instrucciones normales. Un ejemplo sería

```
"\\{" = (
"\\}" = )
"(" = \melisma
")" = \melismaEnd
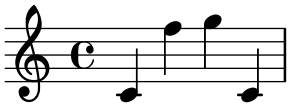
\new Staff <<
  \relative c' {
    c8 \{ d e f \} % slurred
    g ( a b c ) % no slur, but with melisma
    c,1 \bar "|."
   }
   \addlyrics { Li -- ly -- pond. }
>>
```




- El nombre de la abreviatura de articulación para `\staccatissimo` ha cambiado de `-|` a `-!`.

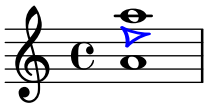
- Ahora se escriben los rangos de cambios de tempo como `\tempo 4 = 60 - 68` en lugar de `\tempo 4 = 60 ~ 68`.
- Se ha cambiado el nombre de `OctavateEight` por el de `ClefModifier`. Las propiedades de contexto relacionadas han cambiado su nombre `xxxOctavationyyy` por el de `xxxTranspositionyyy`.
- Existe una nueva instrucción `\absolute` que marca explícitamente la música introducida en alturas absolutas. Aunque esto era antes el comportamiento predeterminado, una instrucción `\absolute` explícita evita también la reinterpretación cuando el pasaje está dentro de un bloque marcado como `\relative`:

```
\relative c { c'4 \absolute { f'' g'' } c }
```
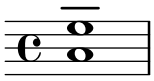


- Cuando se utiliza `\relative` sin emplear como referencia una altura explícita, se toma como referencia el centro de la primera octava, haciendo que no se pueda distinguir si la primera nota que se introduce es relativa o absoluta. Anteriormente, la omisión de una altura de referencia habría llevado a una altura predeterminada de `c'` (el Do central). Dado que la mencionada elección es, hasta cierto punto, arbitraria, la forma de uso recomendada es que se especifique siempre una altura de referencia.
- Se puede usar la nueva instrucción `\single` para convertir la sobreescritura de una propiedad en un truco que se aplica a una expresión musical única:

```
<a \single\voiceTwoStyle e' a>1
```



- Dos maneras de hacer que los objetos gráficos no aparezcan en la salida son: sobreescribir su propiedad `transparent` con un valor verdadero `#t` (reteniendo el espaciado original), o sobreescribiendo su propiedad `stencil` o sello con un valor falso `#f` (que no ocupa ningún espacio). Estas dos operaciones tienen ahora las formas cortas `\hide` (ocultar) y `\omit` (omitir), respectivamente. A estas instrucciones puede dárseles una expresión musical sobre la que efectuar el trucaje, o bien el nombre de un objeto gráfico para el que se debe crear la sobreescritura (para especificar los dos, utilice `\single` sobre la instrucción en forma de sobreescritura):
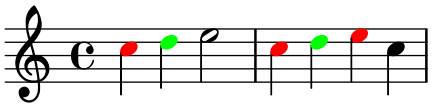
```
\new Staff \with { \omit Clef }
\relative c'' <a e' \hide a>1
```



- Se puede aplicar la nueva instrucción `\temporary` a las sobreescrituras para hacer que no sustituyan a los ajustes de propiedad previos. Si se aplica una instrucción de reversión `\revert` a la misma propiedad varias veces seguidas, el ajuste anterior reaparece:

```
\override NoteHead.color = #red c4
\override NoteHead.color = #green d
\revert NoteHead.color e2
\override NoteHead.color = #red c4
\temporary\override NoteHead.color = #green d
```

```
\revert NoteHead.color e
\revert NoteHead.color c
```



Esto tiene utilidad principalmente para la escritura de funciones musicales que se desea que tengan alguna propiedad modificada solamente por un tiempo igual a la duración de la función.

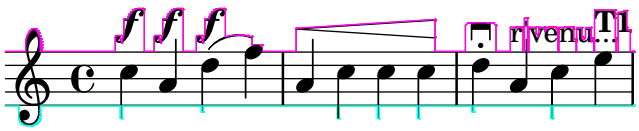
- Las instrucciones `\tag`, `\removeWithTag` y `\keepWithTag` ahora admiten una lista de símbolos en lugar de un solo símbolo para marcar, eliminar y seleccionar música que esté etiquetada con una cualquiera de la lista de etiquetas. Esto es de especial importancia para `\keepWithTag` porque no es posible conseguir el mismo efecto usando varias instrucciones `\keepWithTag` consecutivas.
- La opción '`-d old-relative`' se ha eliminado. Ya no está accesible a través de la línea de órdenes; la utilidad que aún conservaba era la interpretación de `\relative` en archivos de LilyPond convertidos automáticamente de las versiones 1.8 o anteriores. No está claro cuántas de tales versiones estaban aún en funcionamiento.
- Se ha invertido el significado de `instrumentTransposition`. Después de

  ```
  \set instrumentTransposition = #{ b #}
  ```

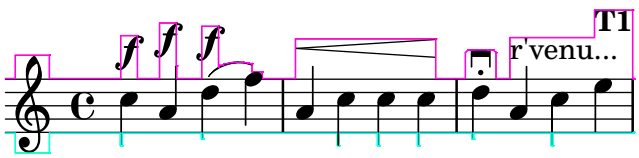
  una nota escrita `c'` ahora suena como `b`. Anteriormente era a la inversa. Esperamos que este cambio y el siguiente hagan más sencillo el tratamiento de los instrumentos transpositores.
- La música generada por las instrucciones `\set` y `\override` ya no resulta afectada por la instrucción `\transpose`. La consecuencia más importante es que `\transpose` transporta la altura tonal sonora o de concierto y la que se imprime en la misma medida incluso aunque la música que se transporta contenga `\transposition`. Anteriormente,

  ```
  \transpose c' f' \transposition bes'
  ```

  era equivalente a `\transposition f'`. Ahora se mantiene como equivalente a `\transposition bes'`.
- Cuando se comprueba la existencia de colisiones, LilyPond ya no trata los objetos como rectángulos. En lugar de ello se aproxima la forma verdadera de los objetos utilizando un enfoque parecido al uso de integrales, lo que por lo común dará como resultado un posicionamiento más ceñido y regular de los objetos y los sistemas de pentagramas:
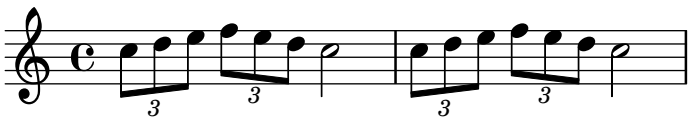



El fragmento anterior tenía anteriormente este aspecto:




Entre los objetos afectados se encuentran `Accidentals`, `Beams`, `Clefs`, `Dynamics`, `FiguredBass`, `Flags`, `Glissandos`, `Lyrics`, `MetronomeMarks`, `OttavaBrackets`, `Pedals`, `RehearsalMarks`, `Rests`, `Scripts`, `TextScripts`, `Ties`, `Tuplets` y `VoltaBrackets`.

- Ahora los grupos de valoración especial se crean mediante la instrucción `\tuplet`, que toma una fracción `t/n` para especificar que se tocan $t$ notas en el tiempo en que habitualmente se permite un número de $n$. Una sola instrucción `\tuplet` puede crear varios grupos especiales si su duración se inserta después de la fracción.

```
\tuplet 3/2 { c8 d e } \tuplet 3/2 { f e d } c2
\tuplet 3/2 4 { c8 d e f e d } c2
```



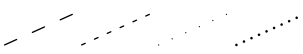
La instrucción `\times` con el orden de la fracción invertido `n/t` aún está disponible.

- Se introducen dos instrucciones nuevas de marcado: `\draw-dashed-line` (trazar línea discontinua) y `\draw-dotted-line` (trazar línea de puntos).

  La línea discontinua se extiende sobre toda la longitud dada por *destino*, si `full-length` está establecido al valor `#t` (que es lo predeterminado) sin ningún espacio al principio ni al final. Entonces, `final` se altera de forma que quepa. Para insistir en los valores dados (o predeterminados) para `inicio` y `final` utilice `\override #'(full-length . #f)`. Son posibles ajustes manuales para `inicio`, `final` y `fase`.

  La línea de puntos siempre se extiende a la longitud completa dada por *destino*, sin ningún espacio al principio ni al final. Es posible ajustar manualmente `final` para obtener espacios mayores o menores entre los puntos. El valor dado (o predeterminado) de `final` se altera de forma que corresponda a la longitud de la línea, `line-length`.
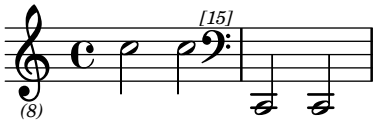
```
\markup {
  \draw-dashed-line #'(5.1 . 2.3)
  \override #'(on . 0.3)
  \override #'(off . 0.5)
  \draw-dashed-line #'(5.1 . 2.3)
  \draw-dotted-line #'(5.1 . 2.3)
  \override #'(thickness . 2)
  \override #'(off . 0.2)
  \draw-dotted-line #'(5.1 . 2.3)
}
```



- A partir de la versión 2.17.10, los mensajes de error o la URI de `textedit` que se usa para la funcionalidad «apuntar y pulsar» especifican los números de columna empezando en 1 en vez de en 0. El desplazamiento de bytes (que también forma parte de las URIs de `textedit`) aún comienza en 0.

- La instrucción `\clef` contempla una transposición opcional:

```
\clef "treble_(8)"
c2 c
\clef "bass^[15]"
c2 c
```



- Se ha hecho que la sintaxis de LilyPond de palabras separadas por puntos `Voice.Accidental` sea intercambiable con `#'(Voice Accidental)`, una lista de Scheme de símbolos. Como resultado, código del estilo de

  ```
  \override Voice.TextSpanner #'(bound-details left text) = "rit."
  ```

  ahora es equivalente a

  ```
  \override Voice.TextSpanner bound-details.left.text = "rit."
  ```

  o incluso a

  ```
  \override #'(Voice TextSpanner) bound-details.left.text = "rit."
  ```

- La ruta a un grob y a una propiedad de un grob ya no requieren que se especifiquen como dos argumentos distintos a instrucciones como '`\override`' y `\revert`, permitiendo la sintaxis siguiente:

  ```
  \override Voice.TextSpanner.bound-details.left.text = "rit."
  ```

  Dado que las funciones musicales complementarias como '`\overrideProperty`' no contemplan formas con y sin espacios de separación al mismo tiempo, la utillización de una sola ruta con puntos es ahora la forma de preferencia. La especificación separada de la ruta de un grob y de la propiedad del grob, contemplada aún por parte de '`\override`' y de '`\revert`' por motivos de compatibilidad, se considera ahora obsoleta.

- Debido a que dos palabras se aceptan ahora como símbolos que son argumentos de función, las interfaces de '`\accidentalStyle`', '`\alterBroken`', '`\footnote`' y '`\tweak`' han tenido que ser rediseñadas allí donde estaban implicados símbolos que eran argumentos opcionales. Compruebe la documentación de la función respectiva para ver los detalles.

- Varias instrucciones aceptan ahora listas de símbolos (introducidas convenientemente como palabras separadas por puntos) para diversos tipos de argumentos. Entre ellos se encuentran '`\accidentalStyle`', '`\alterBroken`', '`\footnote`', '`\hide`', '`\omit`', '`\overrideProperty`', '`\shape`' y '`\tweak`'.

- Se ha modificado la interfaz de usuario de las líneas divisorias. Los glifos de barra de compás ahora se asemejan al aspecto de los caracteres utilizados para expresar el tipo de barra, de manera que un signo de repetición por la izquierda se debe codificar como `.|:`. La instrucción `\defineBarLine` provee una manera sencilla de definir estilos adicionales de línea divisoria.
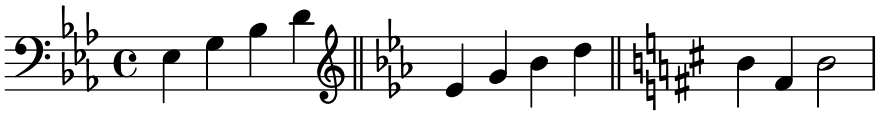
- Las alteraciones accidentales en la armadura de la tonalidad se pueden imprimir en distinta octava de las posiciones tradicionales, o en más de una octava.

